

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE GRADO

Desarrollo de una aplicación Android de tipo quiz

Autor: Daniel Faro García

Tutor: Gonzalo Martínez

Enero-Febrero 2014

La idea de desarrollar una aplicación para dispositivos *Android* de tipo *quiz* surge a partir de dos realidades de la vida cotidiana:

El auge de los móviles como un dispositivo multimedia que ha adquirido una dimensión universal y la de los videojuegos, un fenómeno imparable entre la juventud de hoy en día. De la unión de estos dos puntos, cuya finalidad es generar conocimiento, aparece este proyecto, que intenta de forma ágil, innovadora y divertida dotar a los usuarios de herramientas para aprender por cuenta propia y en cualquier lugar.

El proyecto viene a cubrir un hueco vacío que se ha observado entre otras aplicaciones similares en el mercado actual. Por una parte hay muchas aplicaciones focalizadas en aprender una sola materia, como por ejemplo inglés, de forma repetitiva y monótona. Por otra parte otras sólo se concentran en la parte lúdica, dando prioridad a la diversión sin importar si el usuario de verdad llega a dominar la materia sobre la que se pregunta.

Este proyecto intenta ofrecer una aplicación potente, estable y escalable a muchas materias sin dejar de lado el factor de diversión y el de aprendizaje. La aplicación permitirá tanto repasar al estudiante el día antes de un examen haciendo unos cuantos test, como a un profesor organizar torneos entre los alumnos para intentar provocar esa chispa que haga que estudien como locos para ver quién es el mejor.

El documento que se expone a continuación recoge, además de una puesta en contexto del proyecto, una explicación de las tecnologías utilizadas, un estudio del arte actual, un análisis que recorrerá todas las fases del proyecto desde el diseño hasta las pruebas, y por último un manual de uso.

SUMMARY

The idea of developing a quiz application for Android devices comes from two daily realities:

The boom of mobile as a media device that has acquired an universal dimension, and the rise of videogames, an unstoppable phenomenon among young people. From the union of these points and with the objective to generate knowledge, this project is born to be agile, innovative and fun, giving users the tools they need to learn on their own anytime and anywhere.

The project fills an empty gap that has not been found in other similar applications on the market. On one hand, there are a lot of applications focused on learning a single subject, such as languages, in a repetitive and boring way. On the other hand, others only focus in the playful side, giving priority to fun even if user does not dominate the subject. This project wants to provide a powerful, stable and scalable application without neglecting the fun and learning factor.

The application will allow the student to learn up a subject before an exam doing some tests, whereas the teacher can organize tournaments to cause that spark that makes the students study, so they can figure out who is the best.

The document below has a commissioning project context, an explanation of the technologies that has been used, a study of contemporary art, an analysis that will cover all phases in the project from the design to testing and finally a user manual.

PALABRAS CLAVE

La siguiente lista contiene las palabras más descriptivas que podrían utilizarse para catalogar al proyecto:

- Android
- Juego
- Aprendizaje
- Móvil
- Quiz
- Multijugador

KEYWORDS

The following list contains the most descriptive words which could be used to catalog the project:

- Android
- Game
- Learning
- Mobile
- Quiz
- Multiplayer

ÍNDICE

Resumen.....	i
Summary	ii
Palabras clave.....	iii
Keywords	iii
Índice de figuras	vi
1. Introducción.....	1
2. Contexto de desarrollo	3
3. Estudio del estado del arte	4
3.1 Enfoques existentes	4
3.2 Tecnologías utilizadas	6
3.2.1 Eclipse	6
3.2.2 Android SDK.....	6
3.2.3 PHP	6
3.2.4 POSTGRESQL.....	6
3.2.4 PHPPGADMIN	7
4. Análisis	7
4.1 Requisitos.....	7
4.2 Metodología de desarrollo	8
4.2.1 Contenido de cada prototipo	9
4.3 Aplicación.....	10
4.4 Base de datos	10
4.5 Seguridad	11
4.6 Interfaz.....	11
5. Diseño	12
5.1 Diseño de pantallas	13
5.1.1 Diseño preliminar	13
5.1.2 Diseño del primer prototipo	14

5.1.3	Diseño del producto final	15
5.2	Diseño de la interfaz	16
5.2.1	Diseño del primer prototipo	16
5.2.2	Diseño del segundo prototipo	17
5.2.3	Diseño final	18
5.3	Diseño de la base de datos	19
5.3.1	Diseño preliminar	19
5.3.2	Diseño final	20
5.4	Diseño del PHP	21
5.5	Diseño de la seguridad	21
6.	Desarrollo	21
6.1	General.....	21
6.2	Desarrollo de la aplicación	23
6.3	Desarrollo deL PHP.....	24
7.	Pruebas.....	25
8.	Conclusiones y trabajo futuro.....	26
9.	Anexo Técnico.....	27
9.1	Manual de usuario	27
	Glosario	35
	Referencias	36

ÍNDICE DE FIGURAS

Ilustración 1 Tabla de Moviles por OS	1
Ilustración 2 Aplicación Duolingo	4
Ilustración 3 Aplicación Triviados	5
Ilustración 4 Esquema de prototipados.....	9
Ilustración 5 Trabajo en cada fase	9
Ilustración 6 Diseño de pantallas preliminar	13
Ilustración 7 Diseño de pantallas prototipo1	14
Ilustración 8 Diseño de pantallas producto Final	15
Ilustración 9 Diseño interfaz prototipo1	16
Ilustración 10 Diseño interfaz prototipo 2	17
Ilustración 11 Diseño interfaz final.....	18
Ilustración 12 Diseño esquema bbdd	19
Ilustración 13 Diseño tablas bbdd final	20
Ilustración 14 Desarrollo LocalBroadManager	24
Ilustración 15 Desarrollo consulta ranking.....	24
Ilustración 16 Manual esquema pantallas.....	27
Ilustración 17 Manual menú registro	28
Ilustración 18 Manual menú Login	29
Ilustración 19 Manual menú principal.....	30
Ilustración 20 Manual respuestas correctas.....	31
Ilustración 21 Manual aceptar reto	32
Ilustración 22 Manual estadísticas	33

1. INTRODUCCIÓN

El desarrollo de una aplicación de tipo *quiz* viene motivada por la necesidad de dotar a los alumnos de más herramientas a elegir para el estudio, además de intentar adaptar el modelo de enseñanza actual a uno quizás mucho más interactivo, donde no sólo el docente es parte activa de la docencia sino que el estudiante también puede participar en ella.

Para ello se ha realizado un análisis de las tecnologías actuales disponibles donde se ha encontrado que la tecnología móvil encaja de forma perfecta con esta filosofía. Esto es debido a su universalidad. Según un estudio [1] casi un 80% de los estudiantes universitarios poseen un móvil. Además, éste permite a los alumnos no tener que estar presentes en la facultad o delante de un ordenador.

Una vez elegida la tecnología hay que elegir el sistema operativo, siendo los sistemas predominantes *Android*, *iOS* y *Windows Phone*. Como podemos ver en la siguiente figura:

Top Smartphone Operating Systems, Shipments, and Market Share, 2013 Q3 (Units in Millions)

Operating System	2Q13 Unit Shipments	2Q13 Market Share	2Q12 Unit Shipments	2Q12 Market Share	Year-over-Year Change
Android	187.4	79.3%	108	69.1%	73.5%
iOS	31.2	13.2%	26	16.6%	20.0%
Windows Phone	8.7	3.7%	4.9	3.1%	77.6%
BlackBerry OS	6.8	2.9%	7.7	4.9%	-11.7%
Linux	1.8	0.8%	2.8	1.8%	-35.7%
Symbian	0.5	0.2%	6.5	4.2%	-92.3%
Others	N/A	0.0%	0.3	0.2%	-100.0%
Total	236.4	100.0%	156.2	100.0%	51.3%

Source: IDC Worldwide Mobile Phone Tracker, August 7, 2013

Ilustración 1 Tabla de Móviles por OS

Android parece el candidato perfecto para este proyecto, debido a su gran apoyo por parte del mercado internacional (casi 8 de cada 10 móviles vendidos en el mundo en 2013 son *Android*) y por ser de desarrollo libre, es decir, tanto las

herramientas para hacer aplicaciones como el poder publicarlas en su *Market* es totalmente gratuito (excepto un pequeño cargo por registro de usuario de 25€).

Con este planteamiento inicial se decide desarrollar una aplicación de tipo *quiz* para *Android*, donde los usuarios puedan registrarse y realizar test de las asignaturas que haya en la aplicación. Para ello contará con distintos modos de juego (individual, multijugador...) y distintas formas de autoevaluar su rendimiento en distintas asignaturas (estadísticas, rankings...).

La aplicación tiene tres características sobre las cuales se fundamenta y son: función educadora, diversión/emoción y facilidad de uso y acceso:

- Función educadora: Para lograr este objetivo la aplicación contará con un sistema de progresión, donde nuevos temas se irán abriendo al usuario una vez que haya perfeccionado los temas anteriores. Las preguntas estarán construidas por los propios profesores. Por último los usuarios podrán siempre comprobar las respuestas correctas de todos los test.

- Diversión/Emoción: Es importante y necesario mantener a los usuarios enganchados; para ello el tema de la aplicación se ha hecho colorido y atractivo. Los retos son parte importante, ya que fomentan la rivalidad entre usuarios. También hay implementado un sistema de logros.

- Facilidad de uso: Como se ha comentado anteriormente se ha escogido el sistema operativo *Android* para que el mayor número de usuarios tengan acceso. Además la aplicación ha sido desarrollada teniendo especial cuidado en que funcione en cualquier móvil, ya sea moderno o más antiguo. La interfaz se ha desarrollado con mucha delicadeza ya que es el punto de entrada a la aplicación: es intuitiva y usable.

El documento expuesto a continuación tiene como objetivo detallar la realización y el funcionamiento de la aplicación, de forma que se dividirá la estructura del siguiente modo:

- Una primera parte de introducción y contexto de desarrollo del proyecto donde se explicará qué se ha hecho y para qué.

- Una segunda parte que consiste en la explicación formal de cómo se ha realizado el proyecto, es decir, análisis, diseño y desarrollo. También se hablará de qué tecnologías se han utilizado y se realizará un estudio del arte.

- Una última parte donde se darán ejemplos de funcionamiento, se redactarán las conclusiones y se hablará de posibles ampliaciones futuras del proyecto.

Además de estos últimos puntos, al final del documento se podrá encontrar el glosario de palabras, una lista de referencias y un anexo técnico con un pequeño manual de usuario de la aplicación.

2. CONTEXTO DE DESARROLLO

El contexto de desarrollo se engloba en una asignatura del Grado en Ingeniería Informática llamada Desarrollo de Aplicaciones para Dispositivos Móviles, donde se hacen exámenes semanales de tipo test. En esta asignatura se comprueba que con una cuidada selección de las preguntas se puede conseguir una evaluación más o menos completa de cada alumno, y que con la periodicidad semanal se obliga a éste a estudiar regularmente y así llevar el ritmo de la asignatura. Todo esto desembocó en la siguiente pregunta: ¿Y si usamos las preguntas tipo test para hacerles estudiar, en vez de evaluarlos? Este cambio de idea junto al sentimiento de rivalidad que puede llegar a haber entre alumnos de una misma clase y que les llevaría a hacer test con cierta regularidad desembocó en el desarrollo de este proyecto.

3. ESTUDIO DEL ESTADO DEL ARTE

3.1 ENFOQUES EXISTENTES

En el mundo de las aplicaciones educativas hay dos enfoques claramente diferenciados, aplicaciones más serias donde el objetivo es aprender una materia (ya sea un idioma o cualquier otro tipo de conocimiento), o aplicaciones más lúdicas, donde el único objetivo es la diversión a través de preguntas de ámbito general. Veamos los dos máximos exponentes de ambos tipos de aplicaciones:

-Duolingo: [2]



Ilustración 2 Aplicación Duolingo

Es una aplicación de aprendizaje de idiomas multiplataforma (Web/Móvil/Tabletas) cuyo único objetivo es el aprendizaje del idioma en cuestión aunque con un enfoque menos tradicional. Se vale de preguntas tipo test, alocuciones e imágenes para no volverse repetitivo. Además la aplicación está gamificada (uso de experiencia, vidas, etc.) para conseguir una experiencia de juego más agradable y adictiva.

-Triviados:[3]



Ilustración 3 Aplicación Triviados

Es una aplicación multiplataforma (Web/Móvil/Tabletas) de tipo concurso basada en el Trivial de toda la vida, donde las preguntas tipo test sobre diversos temas son una excusa para jugar en modo multijugador con tus amigos. Apariencia agradable, multijugador y facilidad de uso y acceso son los puntos fuertes de esta aplicación de entretenimiento.

El proyecto se sitúa en un grado intermedio de estos dos puntos de vista, mezclando la intencionalidad de que el usuario aprenda con la diversión y la emoción que proporciona el modo multijugador.

3.2 TECNOLOGÍAS UTILIZADAS

3.2.1 ECLIPSE

Eclipse[4] es un programa de código abierto compuesto por varias herramientas de programación. Se le suelen añadir *plugins* para programar para *Java* (*Java Development Tools, JDT*) o *Android* (*Android Development Tools, ADT*).

3.2.2 ANDROID SDK

El *Android SDK* [5](*Software Development Kit*) es un conjunto de herramientas multiplataforma gratuitas para poder desarrollar para móviles *Android*. Incluye un *debugger*, librerías e incluso emuladores. Estas herramientas se fusionan con *Eclipse* gracias al *plugin ADT* como se ha comentado en el punto anterior.

El punto fuerte del *Android SDK* es la herramienta *Manager*, que está integrada en *Eclipse* gracias al *plugin ADT* y que permite descargar por separado herramientas, documentación o librerías de la versión de *Android* que necesitemos.

3.2.3 PHP

Es un lenguaje de scripting[6] de propósito general que se puede integrar con *HTML* y además es utilizado en servidores para gestionar llamadas *http* y devolver una respuesta. Lo utilizamos para manejar todas las llamadas al servidor y responder.

3.2.4 POSTGRESQL

Es una base de datos relacional de código abierto[7] muy utilizada en entornos *Linux*. Nuestra base de datos, con todos los usuarios y el resto de datos están almacenados en ella.

3.2.4 PHPPGADMIN

Es una aplicación de código abierto web escrita en *PHP* para manejar bases de datos *POSTGRESQL* [8]. Se utiliza para crear o borrar bases de datos, tablas, o lo que sea necesario. También permite ejecutar código *SQL* directamente sobre la base de datos, lo que es muy útil para probar consultas, por ejemplo.

4. ANÁLISIS

Como ya se ha dicho a lo largo de los puntos anteriores, el principal objetivo de esta aplicación *Android* será educar y entretener. Para ello necesitaremos desarrollar con cuidado, fijando unos requisitos que en principio no están muy claros. Por tanto debemos elegir una metodología de desarrollo que realce nuestros puntos fuertes y nos ayude con los puntos débiles.

La arquitectura de la aplicación se ha decidido que tenga el formato de cliente-servidor, siendo el cliente la aplicación móvil. Más concretamente la arquitectura será la siguiente Servidor (Base de datos + *PHP*) – Cliente (Aplicación *Android*).

4.1 REQUISITOS

Lista de requisitos inicial alcanzada tras varias reuniones entre el tutor y el autor de este TFG:

- La aplicación debe ser sobre preguntas y respuestas.
- El número de asignaturas debe ser ampliable.
- El usuario debe poder elegir sobre que asignatura se le preguntara.
- Cada asignatura debe estar dividida en bloques de conocimiento.
- Debe haber un sistema de logros
- Habrá varios tipos de pregunta:
 - Solo Imágenes en Pregunta y Respuesta.
 - Solo Texto en Pregunta y Respuesta.

- Imágenes y texto en Preguntas y Respuestas.
- Tiene que haber diferentes modos de juego:
 - Individual
 - Multijugador
- El usuario debe acumular experiencia.
- La experiencia se ganara al contestar con éxito preguntas.
- Se deben poder ver *Rankings* y algunas estadísticas.
- Debe haber cierto nivel de seguridad en la aplicación.
- Se debe poder dar de alta a usuarios desde la aplicación en la *bbdd*.
- Se podrá *Logear* en el servidor desde la aplicación.
- La aplicación debe ser usable, intuitiva y atractiva.
- La aplicación debe poder estar en varios idiomas.

4.2 METODOLOGÍA DE DESARROLLO

La metodología de desarrollo es el marco que se utiliza para desarrollar un proyecto. Existen varios tipos con sus ventajas e inconvenientes, y el que se escoja depende totalmente del proyecto a realizar. En nuestro caso, los puntos característicos de la aplicación son:

- Primero: aún sabiendo qué queremos que haga la aplicación, los requisitos no están totalmente definidos, por ejemplo, en el primer análisis que se hizo los tipos concretos de modos individuales y multijugador no estaban claros.

- Segundo: la interfaz gráfica es clave, sobretodo en una aplicación móvil. Se necesita que sea intuitiva, usable y agradable. Una vez analizado de qué punto partimos se decide que una metodología basada en prototipos es la que mejor se adapta a este proyecto [9]. Esta metodología se basa en la construcción de aplicaciones que, aunque no son totalmente funcionales, permiten hacerse una idea temprana de cómo será la aplicación final. Concretamente nos basaremos en la construcción de prototipos de características seleccionadas. El primer prototipo tendrá solo cierta parte de la funcionalidad, mientras que los siguientes perfeccionarán la

funcionalidad que ya existía y se añadirán otras nuevas que ya se habían incluido en el proyecto inicial o se haya visto que se necesiten.

Un gráfico a grandes rasgos sería el siguiente:

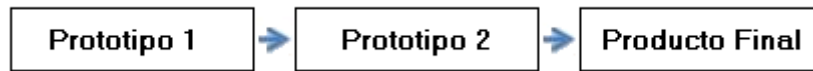


Ilustración 4 Esquema de prototipados

Ahora se mostrará un gráfico más concreto de qué pasos se darán en cada fase:

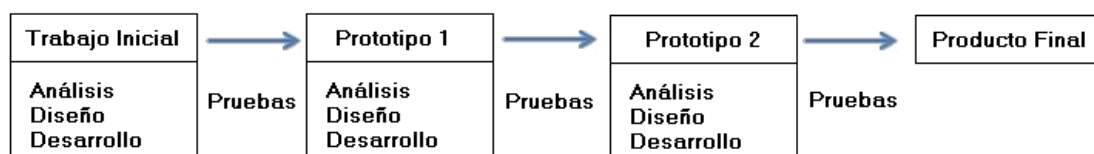


Ilustración 5 Trabajo en cada fase

4.2.1 CONTENIDO DE CADA PROTOTIPO

- En el primer prototipo se creará la base de datos y las primeras pantallas de la aplicación, se podrá registrar un usuario y *loguearse* en la *bbdd*. Además se podrá elegir modos de juego, asignaturas y temas y jugar una partida. Todo con datos estáticos, sin ir al servidor a buscarlos.

- Tras las pruebas del primer prototipo se intentarán refinar todas las interfaces para hacerlas más intuitivas o volver a rehacerlas si no funcionan correctamente. Los datos de las asignaturas y los temas ya se tomarán del servidor, el sistema de retos será ya funcional y se creará una pantalla de puntuaciones.

- El prototipo dos también será probado, y con estos datos se volverá a decidir si se refinan las interfaces o se intenta probar otro diseño. Al producto final se le añadirán las pantallas de ranking y estadísticas, la pantalla que permite ver las respuestas, el sistema para descargar preguntas del servidor y el menú de opciones.

4.3 APLICACIÓN

Se quiere que la aplicación permita dar de alta a usuarios en la base de datos. Habrá partidas individuales y multijugador. Dentro de las individuales habrá dos modos de juego, prácticas y progresión. En el modo prácticas el usuario tendrá acceso libre a todas las asignaturas y a todos los temas contenidos en ellas. En el modo progresión cada partida proporcionará cierta experiencia (puntos) al jugador, y cuando se acumulen suficientes permitirá desbloquear el siguiente tema de la asignatura.

Dentro del multijugador habrá dos modos de juego, uno llamado prácticas, donde dos jugadores contestan preguntas por turnos en un solo dispositivo móvil, y el modo llamado reto en el que dos usuarios compiten entre sí para ver quién domina más cierta asignatura.

La aplicación también deberá mostrar las estadísticas del usuario que está conectado así como los rankings con los mejores usuarios de cada asignatura.

4.4 BASE DE DATOS

Se necesitará una base de datos que dé respuesta a la necesidad de almacenar información entre usos de la aplicación. La primera opción fue crear una base de datos local en el móvil del usuario que almacenara las preguntas y sus propios resultados, pero eso complicaba sobremanera cualquier tipo de juego multijugador. Así que se decidió implementar una base de datos más tradicional, situada en el servidor y que es accedida por unos *scripts* en *PHP*. La base de datos tiene que almacenar tanto la información de cada usuario, como la de las partidas y las preguntas.

4.5 SEGURIDAD

Ya que la seguridad de la aplicación no es un objetivo del proyecto no se va a hacer demasiado hincapié en ella; sin embargo habrá una ligera capa de seguridad para evitar dos casos concretos.

La aplicación se va a comunicar con el servidor por medio de llamadas *HTTP* para recibir y mandar información. Esta información estará a la vista de todos aquellos que consigan interceptarla, como por ejemplo con un ataque de *Man in the Middle* [10]. Si el atacante obtiene las cabeceras que el cliente manda al servidor podría obtener datos muy valiosos como la contraseña o el nombre de usuario. Además podría modificar las cabeceras y enviarlas al servidor para obtener ventajas en el juego.

Esto se puede solucionar de varias formas pero la más rápida y cómoda es cifrar todo el tráfico entre el servidor y el cliente.

La otra capa de seguridad será cifrar las contraseñas de los usuarios para que en caso de que algún atacante consiga acceso a los datos alojados en la base de datos la contraseña no le sea de utilidad.

4.6 INTERFAZ

Para cubrir los apartados anteriores necesitaremos al menos las siguientes pantallas en la aplicación:

- Menú Principal.
- Pantalla para elegir modo de juego.
- Pantalla para elegir asignatura y tema.
- Pantalla para jugar.
- Pantalla para registrarse.

- Pantalla para ver rankings.
- Pantalla para ver estadísticas.

5. DISEÑO

El diseño de la aplicación ha ido variando según se han ido creando prototipos, por lo tanto este apartado irá dividido también para intentar reflejar los pasos que se han ido dando durante el desarrollo del proyecto. Los cambios que se han ido realizando al diseño original surgen por tres motivos:

- Las pruebas del prototipo anterior no fueron satisfactorias en algún modo, como por ejemplo no cubrir algún requisito o existir algún error.
- Faltar algo por añadir que en el primer prototipo no estaba contemplado o se ha pensado de otra forma. Por ejemplo, la experiencia al principio iba a ser del jugador y más adelante pasó a estar dividida por asignaturas.
- Por último, tras adquirir algún conocimiento extra se decidió rehacer alguna parte utilizando un nuevo método.

5.1 DISEÑO DE PANTALLAS

5.1.1 DISEÑO PRELIMINAR

En este diseño preliminar aún no estaba muy claro cómo iba a funcionar toda la parte del juego en sí, por eso no está totalmente definido. El resto de la aplicación no ha cambiado de forma substancial a lo largo del resto de iteraciones del proyecto.

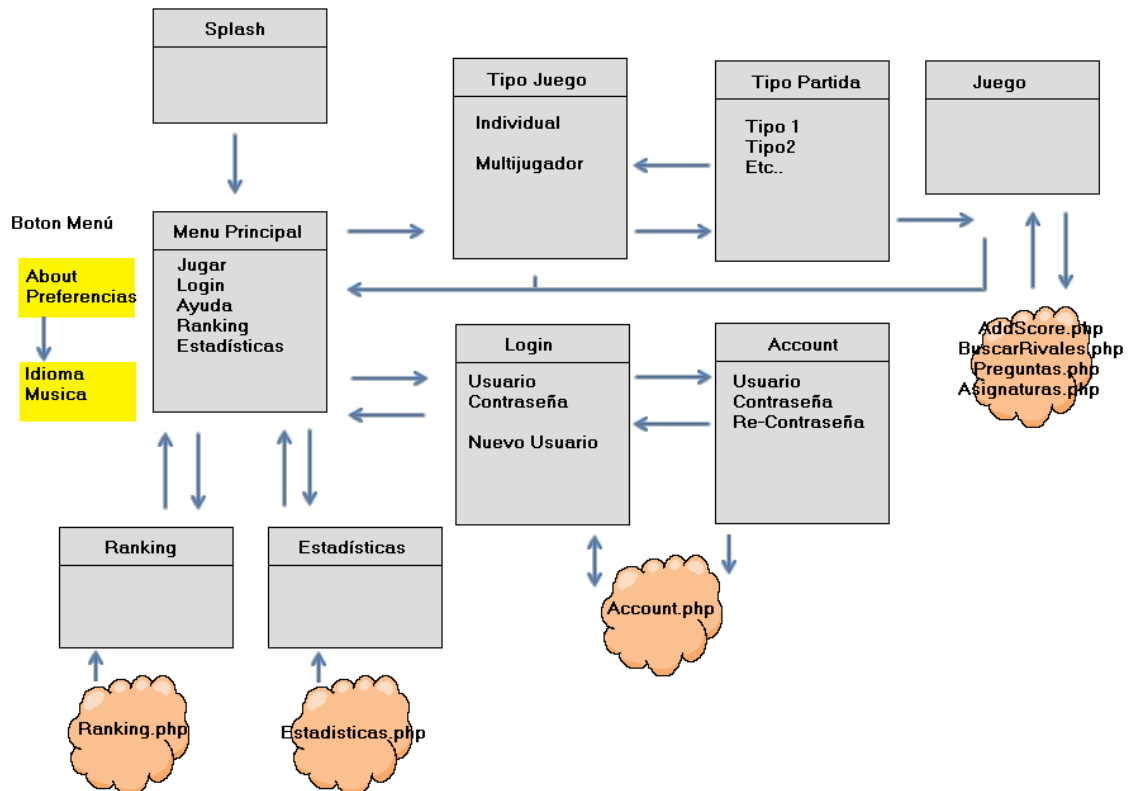


Ilustración 6 Diseño de pantallas preliminar

5.1.2 DISEÑO PRIMER PROTOTIPO

En este diseño de pantallas se puede comprobar que la parte del juego ya ha adquirido forma, se sabe cómo se va a dividir y cómo interactúa cada pantalla con el servidor. Las preguntas son descargadas desde la pantalla de juego y las asignaturas desde el menú principal y almacenado, ya que es algo que va a ser utilizado por casi toda la aplicación y no son datos que van a ser modificados con frecuencia. El botón de menú se ha dejado de utilizar y ahora las opciones aparecen en el *ActionBar*.

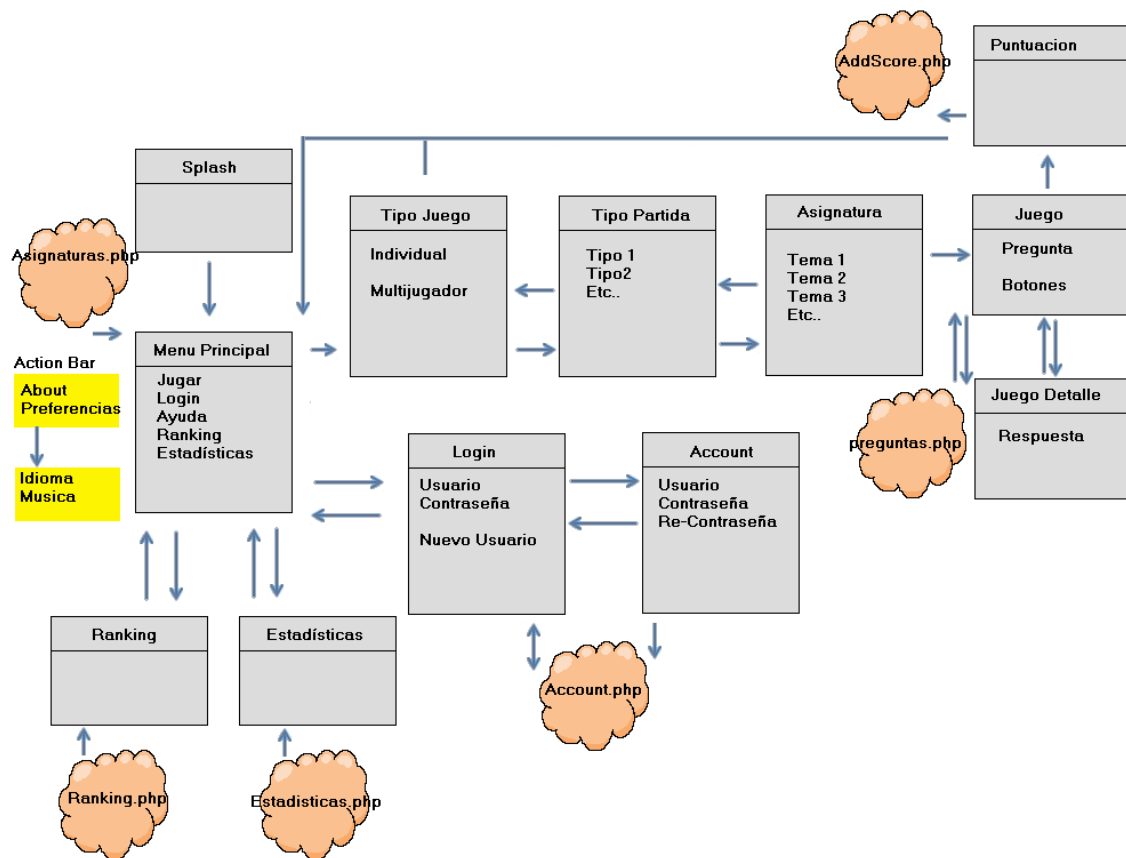


Ilustración 7 Diseño de pantallas prototipo1

5.1.3 DISEÑO PRODUCTO FINAL

En esta imagen ya se ve como se ha definido toda la parte del los retos, con el comprobador en el menú principal y “solicitar reto” tras la pantalla de asignaturas. El funcionamiento del juego ha vuelto a cambiar para ser más atractivo y se ha añadido una pantalla de respuestas correctas.

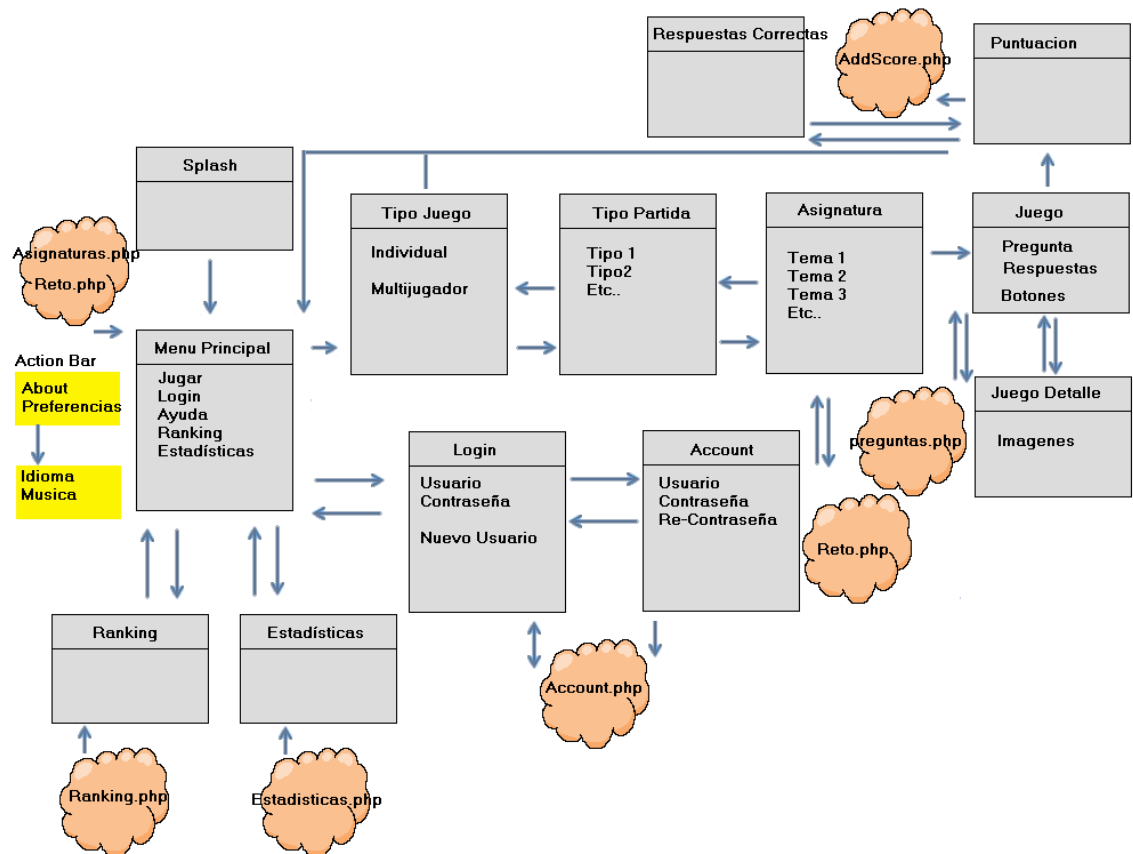


Ilustración 8 Diseño de pantallas producto Final

5.2 DISEÑO DE LA INTERFAZ

5.2.1 DISEÑO PRIMER PROTOTIPO

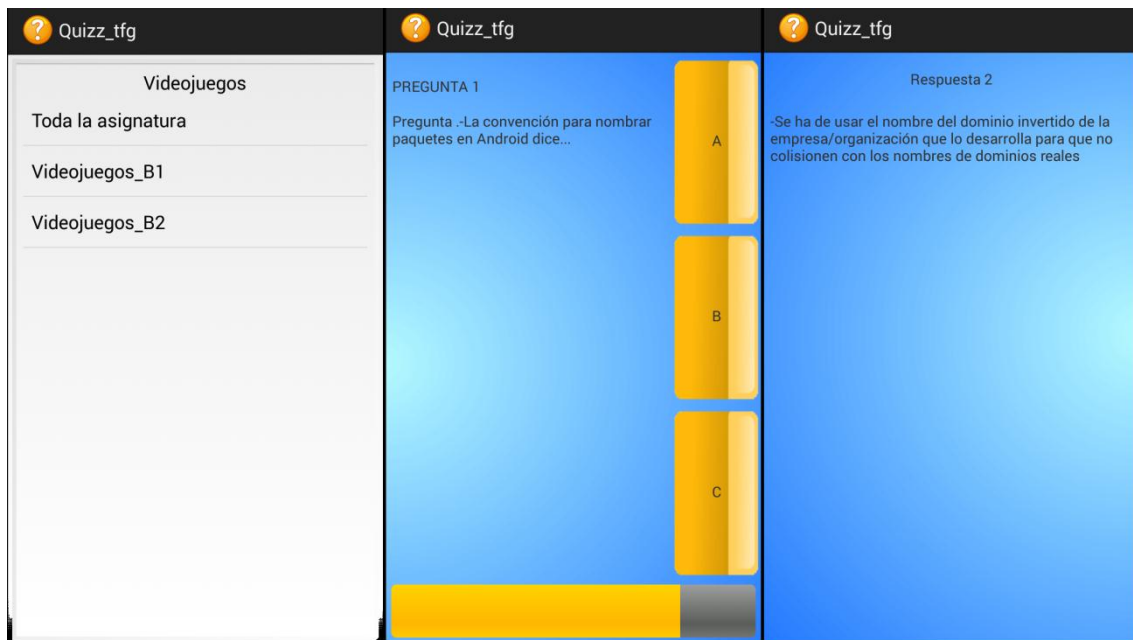


Ilustración 9 Diseño interfaz prototipo1

Este modelo de interfaz tenía un gran problema: no era intuitiva. En la pantalla de selección no había nada que indicase que había que arrastrar para poder pasar de asignatura. Esto hacía a los probadores dudar y finalmente sólo escogían temas de la primera asignatura.

Con la pantalla de partida nos encontrábamos varios problemas. Era casi imposible que ninguna pregunta tuviera tanto texto como para rellenar todo el hueco y por tanto era un desperdicio de espacio. El funcionamiento de los botones no era natural, la gente no entendía que el pulsado largo era para mostrar la pregunta y llevaba a equívocos.

5.2.2 DISEÑO SEGUNDO PROTOTIPO

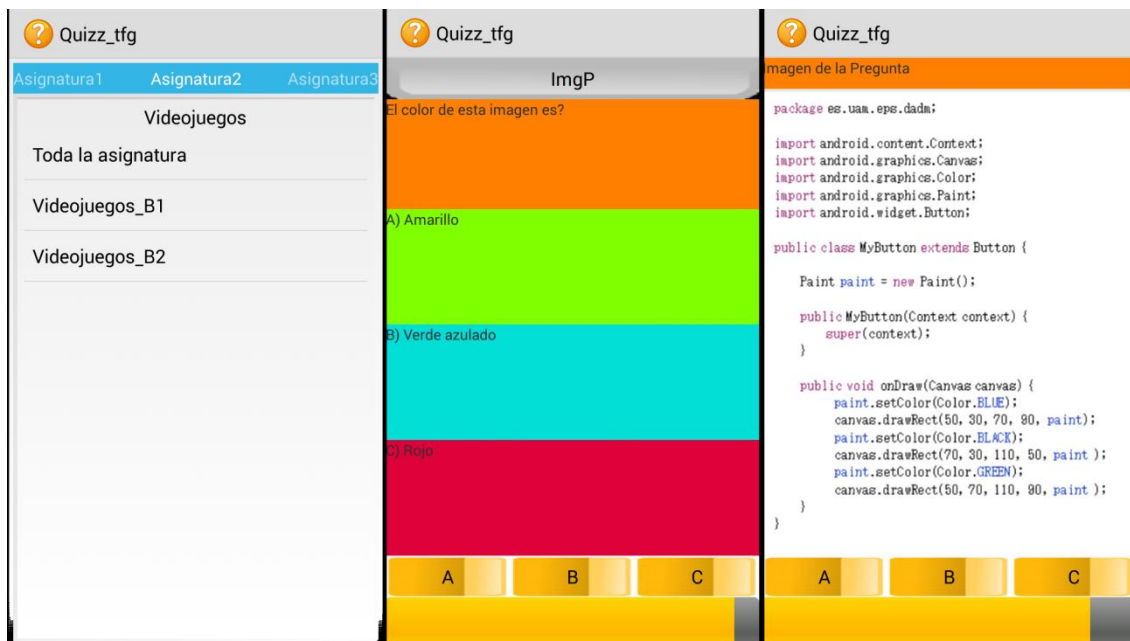


Ilustración 10 Diseño interfaz prototipo 2

Esta vez para evitar el pulsado largo sobre los botones y para aprovechar todo ese espacio bajo la pregunta se intentó colocar todo en la misma pantalla. Se podía tocar cada división de texto para aumentar esa parte y que cupiera todo el texto. Tras realizar pruebas, este formato tampoco parecía funcionar. Los probadores no sabían cómo agrandar las preguntas que no cabían y además, por cómo estaba construida la interfaz la única parte donde se podía arrastrar para pasar la pregunta era la barra de tiempo lo cual llevaba aún a más equívocos.

5.2.3 DISEÑO FINAL

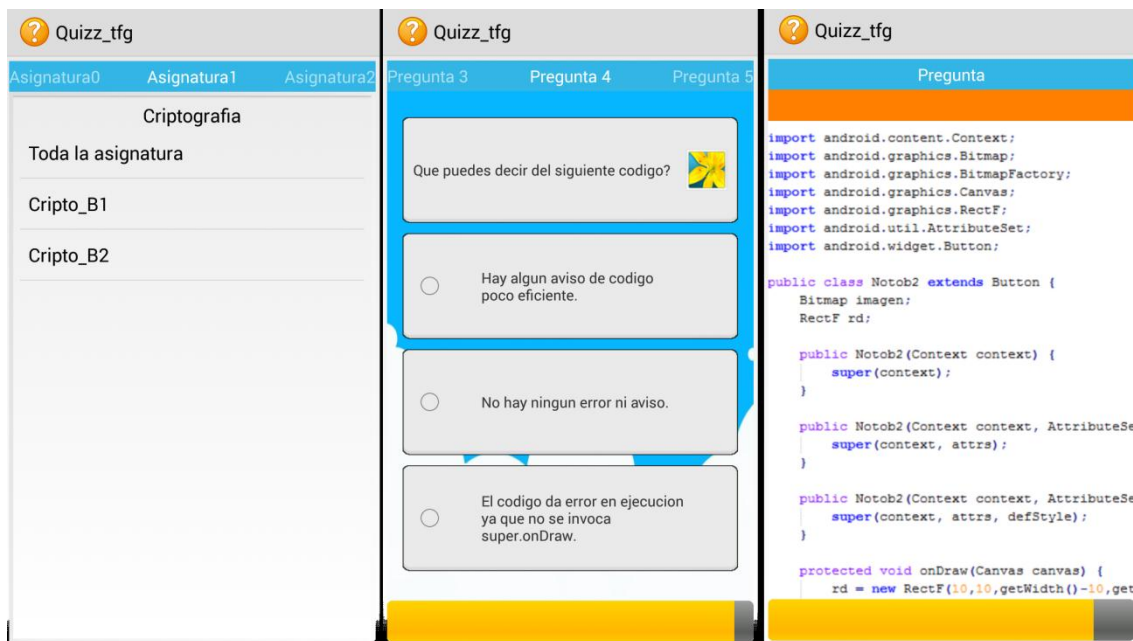


Ilustración 11 Diseño interfaz final

Un análisis de los datos proporcionados por los probadores y un amplio estudio sobre construcción de interfaces en *Android* desembocó en esta última interfaz, la cual al ser probada funcionó de la forma esperada. La barra superior muestra que hay más preguntas delante y detrás, eso consigue hacer mucho más intuitiva la aplicación, y el uso del scroll en todos los textos hace que por muy grande que sea el texto no desencuadre ningún elemento de la interfaz.

5.3.1 DISEÑO PRELIMINAR

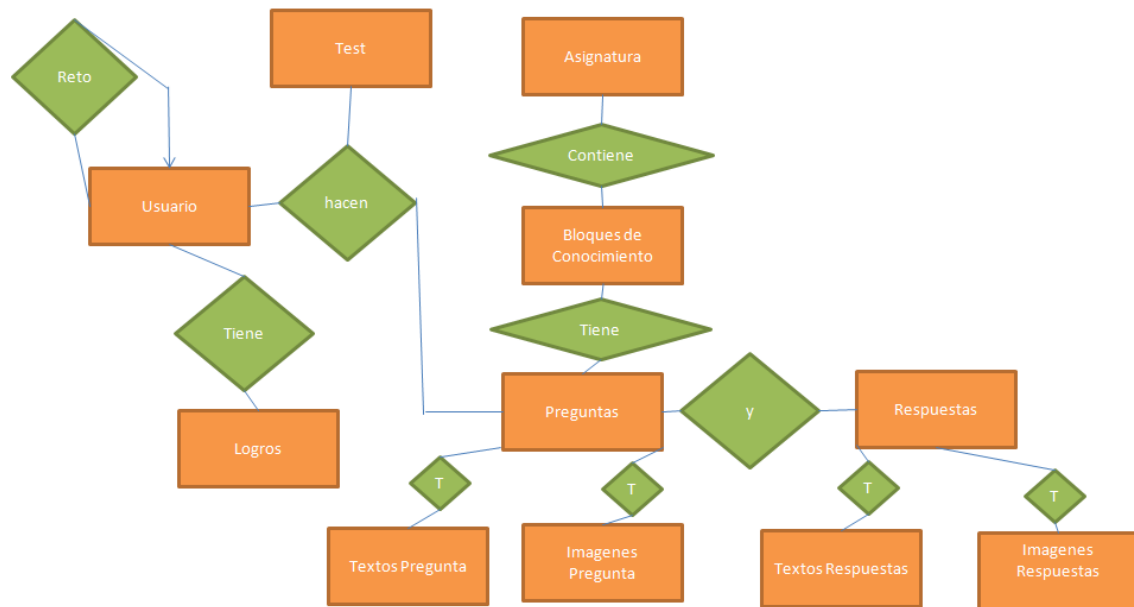


Ilustración 12 Diseño esquema bddd

Esta imagen muestra el primer esbozo del esquema entidad relación de la base de datos que se hizo durante el análisis, en el se puede observar como no había tablas dedicadas a almacenar la experiencia ni el ELO de cada usuario, por lo tanto eran comunes a todas las asignaturas. Tampoco se tenía en cuenta el que las preguntas podían estar en distintos idiomas. Es incorrecto en muchos aspectos pero muestra cómo el proyecto ha ido evolucionando.

5.3.2 DISEÑO FINAL

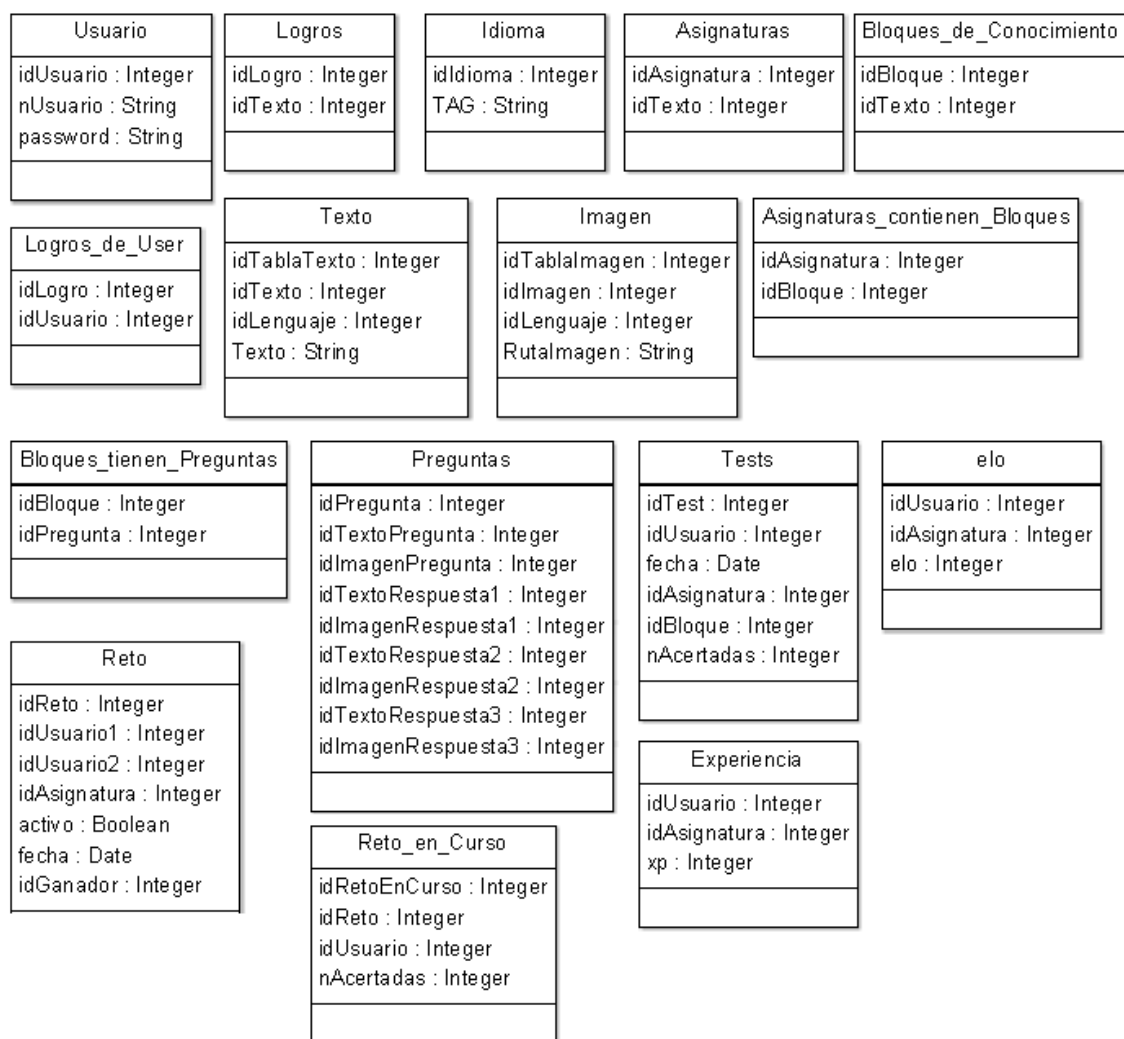


Ilustración 13 Diseño tablas bddd final

En esta imagen se ven todas las tablas de la base de datos con todos sus atributos. Todo lo relacionado con preguntas ha sido llevado a una sola tabla lo cual hace su manejo y las consultas resultantes mucho más fáciles. Las tablas Texto e Imágenes son necesarias debido a la posibilidad de varios idiomas, por ejemplo, para el mismo idTexto de una asignatura, habrá tantas entradas en Texto como Idiomas haya, es decir, estará totalmente traducido. La tabla reto en curso existe debido a la necesidad de almacenar las partidas de los retos que aún no han terminado.

5.4 DISEÑO DEL PHP

Respecto al *PHP* ha habido que tomar pocas decisiones de diseño. Existe un *script* por cada acción importante de la aplicación (*ranking*, *account*, estadísticas, preguntas, asignaturas...) y todos se comportan de la misma manera: reciben una serie de parámetros cifrados, los cuales se decodifican y se trabaja con ellos y cuando hay que devolver una respuesta se hace o bien a través de un código de retorno o devolviendo un documento *XML* con todos los datos solicitados.

5.5 DISEÑO DE LA SEGURIDAD

Se ha decidido que el sistema de cifrado de las contraseñas sea *Sha256 + Salt*. Con esto se consigue que aunque alguien consiga acceso a la *bbdd* y robe las contraseñas, y aun sabiendo qué tipo de cifrado se está utilizando, le sea más complicado conseguir la contraseña sin cifrar gracias al *Salt*[11]. Tendría que recurrir al uso de una fuerza bruta muy costosa o al uso de *rainbow tables*[12].

6. DESARROLLO

En este apartado de desarrollo se van a comentar sólo los puntos más complicados, interesantes o que necesiten explicaciones:

6.1 GENERAL

Para cumplir con el requisito de que la aplicación funcione en todos los móviles posibles independientemente de su antigüedad ha habido que utilizar las librerías de soporte de Google. Estas librerías de soporte dotan a las versiones más antiguas de *Android* de elementos que han sido introducidas en las modernas. Un ejemplo del uso de estas librerías ha sido todo el trabado con fragmentos [13]. Un fragmento es una porción de la interfaz de una Actividad; éstas se pueden combinar, modificar, quitar y poner a voluntad. En nuestra aplicación se han utilizado para modificar partes de la pantalla de forma fácil y limpia. La parte incómoda de los fragmentos es la comunicación entre ellos y su actividad padre, pues tiene que ser hecha a través de funciones *callback*.

Los *ViewPager*[14] que se han utilizado para manejar los *slides* también han necesitado de esta librería de soporte puesto que realmente son una implementación de fragmentos. El *ViewPager* es una *FragmentActivity* a la cual se le pasa un *PagerAdapter* que puede contener cualquier cosa. En uno de nuestros casos, por ejemplo, es un *ListAdapter*, y el *ViewPager* envuelve esa lista en un fragmento de forma automática y cada vez que pasas de página lo modifica y carga otra lista.

Para poder utilizar la barra de acción (*ActionBar*) [15] y así no tener que utilizar el botón Menú de los móviles (cuyo uso ya no está apoyado por Google [16] "*Android-powered devices are no longer required to provide a dedicated Menu button. With this change, Android apps should migrate away*") ha sido necesario recurrir a un proyecto de código abierto [17] ya que la librería de soporte de Google no alcanza hasta modelos tan antiguos como necesitamos.

Al principio las imágenes iban a estar guardadas dentro de la propia base de datos como arrays de bits, pero debido al tamaño de las imágenes se decidió que esa solución no iba a ser funcional. Una solución temporal que se ha encontrado es la de almacenar las imágenes en el propio proyecto. Esto hace que la velocidad al obtenerlas sea instantánea pero dificulta todo lo que tiene que ver con su modificación y los *updates* de la aplicación una vez que está desplegada, puesto que para añadir nuevas imágenes los usuarios tendrán que actualizar la aplicación.

Un punto interesante del desarrollo es toda la comunicación entre la aplicación y el servidor. Para realizar esta tarea contamos con dos clases *Service* y *AsyncTask*, ambas funcionan sin interacción por parte del usuario. Un *Service* se lanza desde una Actividad pero no tiene porqué terminar cuando se termine la actividad, puede continuar funcionando. El *Service* funciona en el hilo de la interfaz gráfica. Esto es un problema, ya que si el servicio se bloquea o consume muchos recursos de la CPU el usuario experimentara bloqueos, botones que no se pulsan etc. y esto es inaceptable.

Por otra parte una *AsyncTask* está totalmente ligada a una Actividad, es decir su ciclo de vida empieza y acaba a la vez que la Actividad que lo lanzó. En cambio se ejecuta en un hilo diferente al de la IU, lo cual evita problemas de bloqueos como ocurre con los *Services*.

Como nuestra aplicación necesita comunicarse con el servidor y recoger datos, la estrategia que se ha seguido es la de envolver tareas asíncronas en servicios, así se consigue que no estén ligados a la actividad que lo lanzó y que no vayan en el hilo de la IU.

Esta solución presenta otros retos por sí misma. Como la tarea asíncrona no está en el hilo de la IU, dependiendo de qué tenga que hacer la tarea se puede necesitar proveer una forma de comunicación entre ambos hilos. Esto se ha solucionado de dos formas diferentes, con el envío de mensajes y con el uso del *LocalBroadcastManager* más el envío de *Intents*. Básicamente la idea es que el hilo de la IU tenga una función que capture todos los *Intents* que son lanzados desde la tarea asíncrona, y una vez tenga el *Intent* puede obtener los datos que estaban guardados en él y trabajar con ellos.

```

/*Juego:OnCreate*/
LocalBroadcastManager.getInstance(this).registerReceiver(
    mMessageReceiver, new IntentFilter("cargaTerminada"));

/*Juego*/
private BroadcastReceiver mMessageReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        crearPager();
    }
};

/*Servicio*/
private void sendLocationBroadcast(Intent intent){
    LocalBroadcastManager.getInstance(this).sendBroadcast(intent);
}

```

Ilustración 14 Desarrollo LocalBroadManager

6.3 DESARROLLO DEL PHP

Otro punto interesante es la formación de los rankings. Para poder mostrar las estadísticas se necesita extraer la experiencia, la posición en el ranking de *ELO*, el *ELO* y el *idAsignatura* del usuario en todas las asignaturas de la base de datos.

La consulta que se realiza para obtener dicha información es la siguiente:

```

SELECT
    ranking.ranks as posicion,
    ranking.elo as elo,
    experiencia.xp as experiencia,
    ranking.idAsignatura as idAsignatura
FROM
    (SELECT
        Rank() over (Partition by idAsignatura Order By elo DESC) as Ranks,
        idAsignatura,
        idUsuario,
        elo
    FROM
        elo
    GROUP by IdUsuario,idAsignatura,elo
    Order by ranks ASC) ranking
LEFT OUTER JOIN experiencia on (ranking.idUsuario=experiencia.idUsuario and
                                ranking.idAsignatura=experiencia.idAsignatura)

```

Ilustración 15 Desarrollo consulta ranking

Se puede observar que hay dos consultas, una dentro de otra. La interior es la encargada de crear los rankings de todos los usuarios por cada asignatura. Funciona

gracias a la función de *SQL RANK*, encargada de crear Rankings según la columna que le digas (Asignatura en nuestro caso). Con el resultado de esa primera *Query* se hace un *Left Outer Join* con la tabla experiencia. Esto se hace así porque a los datos que ya tenemos necesitamos añadirle la columna *exp* de la tabla experiencia pero el problema aquí es que puede que en la tabla experiencia no esté ese usuario para esa asignatura. Es decir, que no tenga experiencia el *Outer Join* soluciona eso concretamente, si en la tabla con la que va a hacer el *Join* no está esa fila, coloca un *null*. Por último, después de hacer el *Join* nos quedamos con las filas con las que salga el usuario.

7. PRUEBAS

Las pruebas han sido muy sencillas de preparar y realizar. Tras terminar cada prototipo las pruebas realizadas han sido las mismas. Se han dividido en tres tipos, pruebas de caja blanca, pruebas de caja negra por pantallas y pruebas con probadores.

- Pruebas de caja negra:

Se han escogido unas cuantas funciones de la aplicación que se sabe que podrían dar problemas por su mayor nivel de complejidad, como pueden ser todos los servicios y tareas asíncronas, y se han probado paso a paso, así como todos los *Scripts PHP*. Así se solucionaban de forma rápida problemas graves de comunicación entre servidor y aplicación que la dejarían sin funcionamiento.

- Pruebas de caja blanca por pantallas:

Son pruebas de integración donde han sido probadas todas las pantallas. La forma de probarlas ha sido la siguiente: comprobar que toda la funcionalidad que ofrece la pantalla funciona correctamente y que está bien integrada con el resto de la aplicación y con la base de datos. Para ello siempre se comprobaba que el resultado que debía quedar en la base de datos era el correcto.

- Pruebas con probadores:

Una vez se tenía un prototipo finalizado las últimas pruebas eran las mismas, ofrecerle la aplicación a gente para que la utilice como si fuera el usuario final.

8. CONCLUSIONES Y TRABAJO FUTURO

La aplicación funciona correctamente, es estable y cumple sus cometidos. Espero que se empiece a utilizar como herramienta de apoyo y aprendizaje en las asignaturas de las carreras (aunque su uso no se vea limitado al ámbito universitario).

A la aplicación aún le queda mucho por añadir y mejorar, como puede ser añadir nuevos modos de juego o nuevos tipos de preguntas (como puede ser rellenar un hueco con texto). Se puede además crear una aplicación de escritorio o web que ayude a agilizar el proceso de carga de preguntas a la base de datos que, aunque es fácil es muy tedioso. Otras posibilidades es distribuirlo a través del *Market* de Google para que todo el mundo tenga acceso a la aplicación o incluso implementarla para otros sistemas, como podría ser *iOS* o *Windows Phone*.

Otro objetivo futuro es solucionar el problema del almacenaje de las imágenes de las preguntas. Una solución válida sería almacenarlas en un servidor *FTP* y que la primera vez que el la aplicación se conectase se descargara todas las imágenes que no tuviera ya guardadas.

9. ANEXO TÉCNICO

9.1 MANUAL DE USUARIO

El manual de usuario de la aplicación tendrá el siguiente formato: nombrará una acción que se puede realizar, como por ejemplo, "jugar un reto" y a continuación se describirán detalladamente todos los pasos necesarios para llevarla a cabo. Por último se explicarán conceptos como *ELO*, experiencia y cómo funcionan los retos. Antes de nada se mostrará un esquema con las pantallas que tiene la aplicación para que el usuario lo tenga presente:

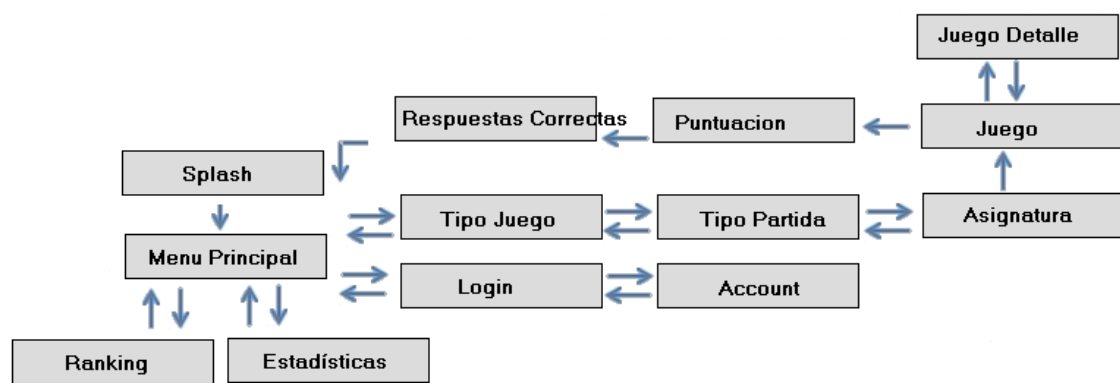


Ilustración 16 Manual esquema pantallas

- Registrarse:

En la pantalla del menú principal y sin estar *Logueado* aparecerá la opción de *Login*. Una vez pulsada se abrirá la pantalla de *Login*, siendo uno de los botones *Nuevo usuario*, que al pulsarlo llevará a la pantalla *Cuenta* donde tras introducir los datos (nombre de usuario y contraseña) se quedará registrado.

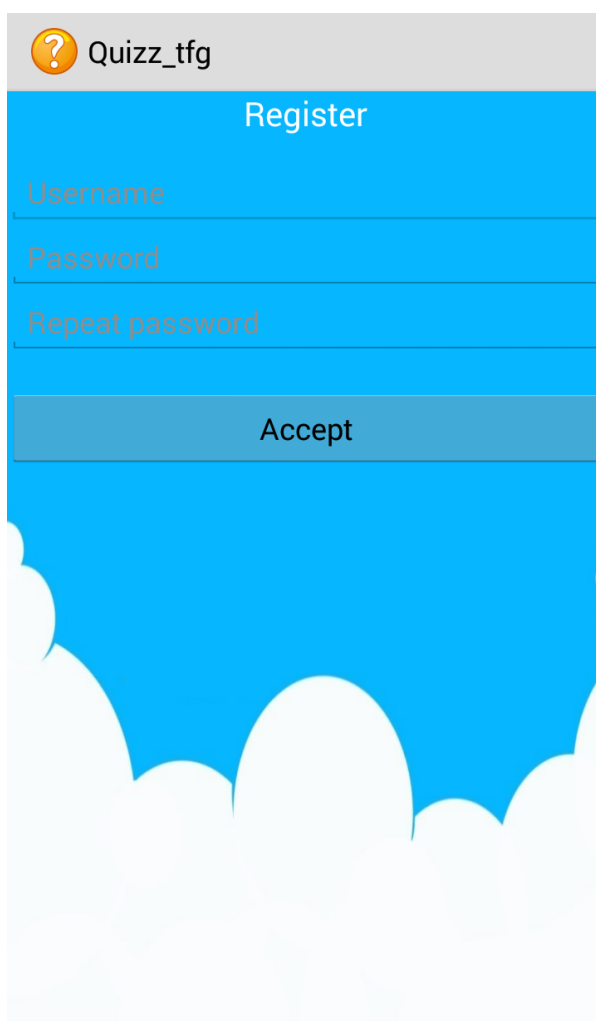


Ilustración 17 Manual menú registro

- Loguear/Desloguear:

En la pantalla del menú principal y sin estar *Logueado* aparecerá la opción de *Login*. Una vez pulsada se abrirá la pantalla de *Login*, al introducir en ellas su nombre de usuario y contraseña y pulsar *Aceptar* se validarán y quedará así *Logueado*.

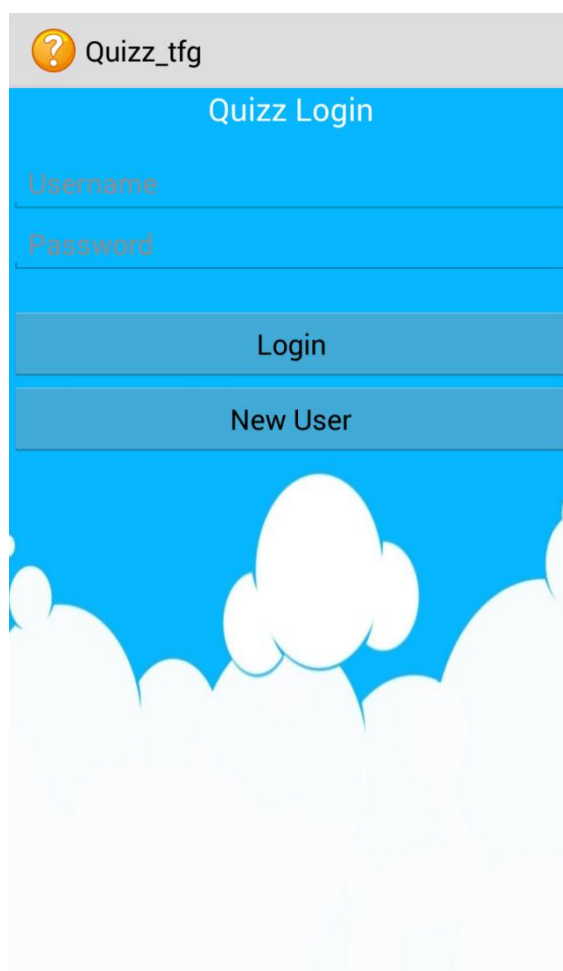


Ilustración 18 Manual menú Login

Para *Desloguear* estando en la pantalla principal, en la barra superior se encuentra el botón para llevar a cabo esta acción.

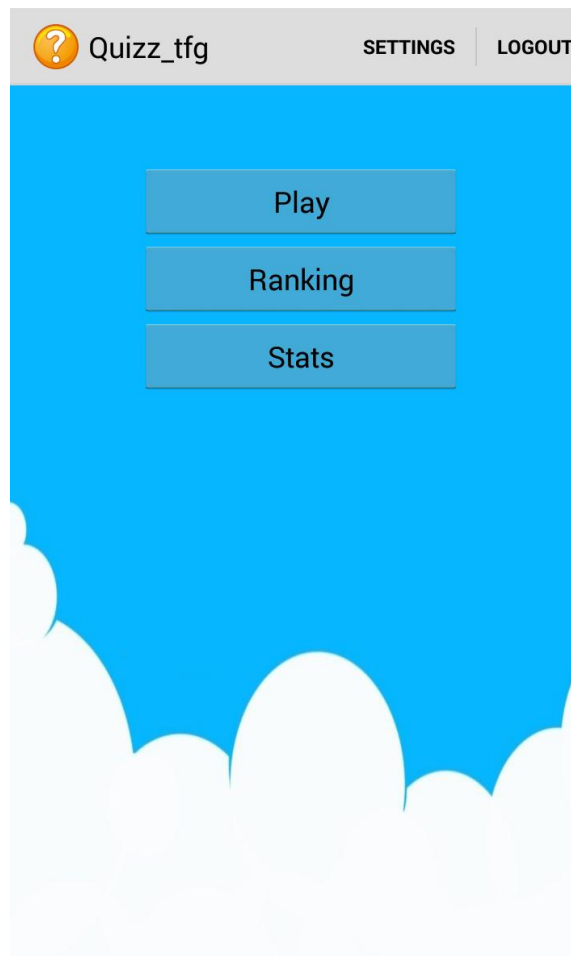


Ilustración 19 Manual menú principal

- Jugar:

Estando *logueado* y desde el menú principal se escogerá la opción de Jugar, se dará a elegir el modo de juego y la asignatura. Una vez empiece la partida se observarán varios elementos. Abajo del todo un contador de tiempo, que cuando se vacíe se acabará la partida. La pregunta tendrá un texto y si tiene una imagen asociada tendrá un botón a su derecha. Las respuestas tienen un botón redondo para poder escoger la respuesta acertada, a continuación viene el texto de la pregunta y por último puede que haya un botón para mostrar una imagen. Para pasar de pregunta solo hay que hacer un arrastre lateral con el dedo (*slide*). Si se pulsa el botón de imagen se abrirá otra pantalla donde se mostrará (aquí podemos realizar la acción de

zoom, también podemos realizar la acción de *slide* para ver otras imágenes). Pulsar el botón de atrás del móvil nos llevará de nuevo a la pantalla de selección de respuestas. Una vez hayamos terminado con las preguntas llegaremos a un menú que nos mostrará el número de preguntas acertadas, y además nos preguntará si queremos volver a jugar o volver al menú principal. Dependiendo del modo de juego escogido también puede aparecer la opción de ver las respuestas correctas.

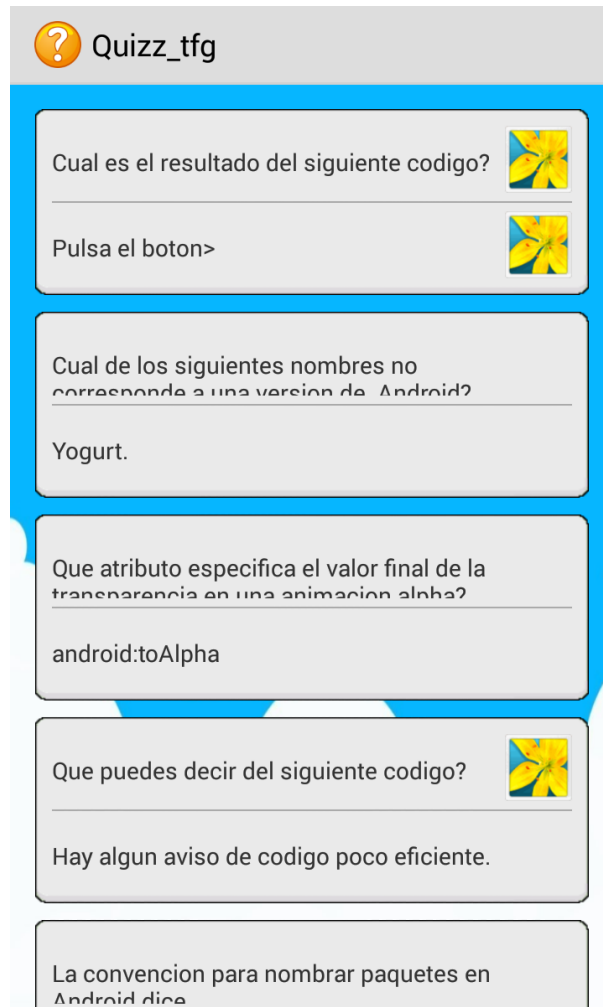


Ilustración 20 Manual respuestas correctas

- Solicitar un Reto:

En el menú principal, al pulsar el botón de *Jugar*, el de *Multijugador* y a continuación *Reto*, ahora se debe escoger la materia sobre la que se realizarán las preguntas. Después de escogerla se devolverá al usuario al menú principal.

- Aceptar un Reto:

Si se está en el menú principal siempre se comprueba si hay retos disponibles, es decir, si has solicitado un reto de cierta asignatura y otro usuario también lo ha solicitado, aparecerá una pantalla preguntando si quieres comenzar el reto. Al pulsar *Sí* se llevará al usuario a la pantalla de juego. El usuario del reto que conteste más preguntas ganará la partida.

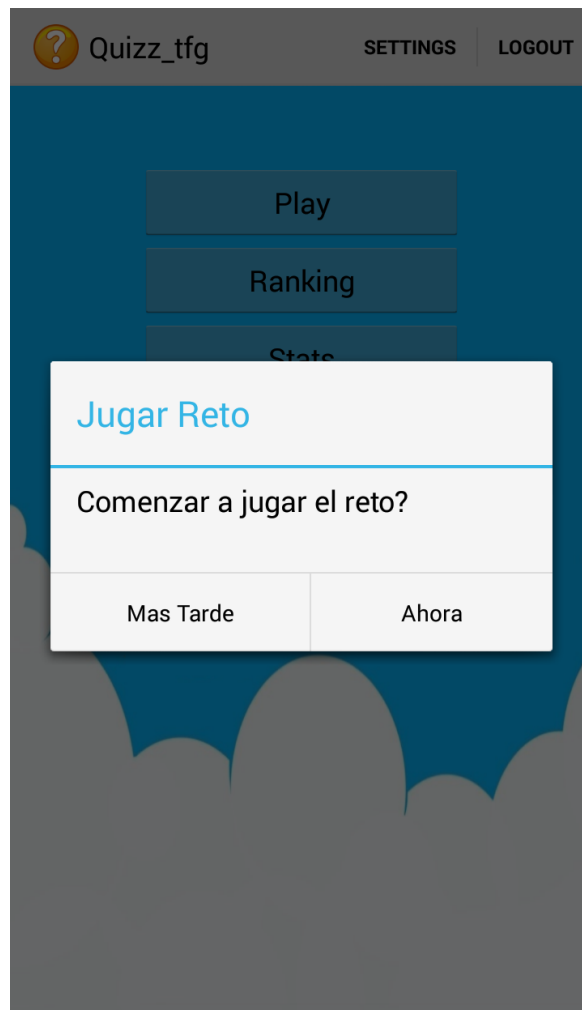


Ilustración 21 Manual aceptar reto

- Ver Estadísticas o Rankings:

En el menú principal, pulsar el botón de *Estadísticas o rankings*. En estos menús hay campos selectores de respuestas para poder modificar la información que se está mostrando. Por ejemplo para ver el *Ranking* de experiencia de los usuarios de la Asignatura X solo hay que seleccionarlo en los selectores.



Ilustración 22 Manual estadísticas

- Cambiar el idioma de la aplicación:

No se puede cambiar directamente desde la aplicación, sino que ésta funciona detectando el idioma de tu teléfono móvil y adaptándose. En caso de que el idioma no esté soportado por la aplicación, el idioma por defecto es el español.

- Ir al menú de opciones:

En la pantalla del menú principal y estando *Logueado*, pulsar el botón de opciones de la barra de acción superior.

- ELO:

El sistema de puntuación *ELO*[18] es muy utilizado para calcular niveles de habilidad relativas entre usuarios. En esta aplicación se utiliza en los retos para poder crear un ranking. Gracias a este sistema se intenta emparejar siempre a usuarios de más o menos el mismo nivel de habilidad.

- Reto:

Cuando se solicita un reto se comprueban todos los retos de los usuarios que aún no se han iniciado; si hay alguno que esté más o menos al mismo nivel de habilidad se le seleccionará como rival. En caso de que la diferencia del usuario que reta con los que están sin jugar sea muy grande se le añadirá a la cola a la espera de un retador de su mismo nivel. Una vez han seleccionado ambos jugadores el reto, se puede empezar a jugar la partida. Esta acción no tiene que ser a la vez, sino que cuando uno se conecte tendrá la oportunidad de jugar si quiere. Una vez ambos jugadores hayan jugado, se determinará el ganador y se repartirán los puntos *ELO*.

- Experiencia:

La experiencia está directamente relacionada con el modo *Progresión*. Está dividida por asignaturas, es decir la experiencia del usuario A en la *Asignatura X* no es la misma que la que tiene en la *Asignatura Y*. Se consigue al contestar acertadamente preguntas en cualquier modo que no sea de prácticas. Por cada 500 puntos de experiencia que consiga el jugador en una asignatura se desbloqueará un

tema nuevo del modo *Progresión*. Para comprobar la experiencia que se tiene solo hay que ir al menú de estadísticas y seleccionar la asignatura que se desee consultar.

GLOSARIO

A continuación podemos ver una lista de palabras cuyo significado puede ser desconocido o de difícil interpretación para el lector:

- **ActionBar:** En *Android*, es una barra que aparece en la parte superior de la ventana con iconos de acciones predefinidas en la aplicación, como puede ser Buscar.
- **Activity:** En *Android*, una actividad es una clase encargada de crear una ventana en la cual poner tu interfaz de usuario, tienen su propio ciclo de vida que suele empezar con una llamada a la función *onCreate()* y acaba (normalmente) con una llamada a la función *onDestroy()*.
- **Debugger:** También llamado depurador, es un programa utilizado para probar y depurar otros programas.
- **Frament:** En *Android*, representa el comportamiento de una porción de la interfaz de usuario, se utiliza para crear una interfaz de usuario modular. Cada fragmento tiene su propio ciclo de vida.
- **Fragment Activity:** En *Android*, es una clase descendiente de *Activity* que soporta el uso de fragmentos. Se puede llamar desde la librería de soporte para que móviles más antiguos tengan acceso a ella.
- **Gamificar:** El uso de mecánicas de juego fuera del contexto de los mismos con el fin de que las personas adopten cierto comportamiento, por ejemplo, añadir el uso de puntos a una aplicación de un entorno de trabajo corporativo.
- **Intent:** En *Android*, es una descripción abstracta de una operación que tiene que ser realizada.
- **Interfaz (de usuario):** Es el medio con el que un usuario interactúa con una aplicación.
- **Left Outer Join:** Es un tipo especial de *Join*, es una operación que une dos tablas. El resultado será toda la tabla de la izquierda más las columnas de la tabla derecha que cumplan ciertos parámetros.
- **Loguear:** Acto de identificarse para obtener acceso a un sitio restringido.
- **Market:** En el ámbito de los teléfonos móviles, el *market* es el lugar donde se descargan y venden tanto aplicaciones como todo tipo de software especializado para móviles.
- **Multijugador:** Entorno donde interactúan varios jugadores entre sí.
- **Multiplataforma:** Entorno que puede funcionar en distintos sistemas operativos.

- **Query:** En el ámbito de las bases de datos, es realizar una consulta con el objetivo de conseguir ciertos datos.
- **Quiz:** Un tipo de juego de concurso donde los jugadores (de forma individual o por equipos) intentan responder de forma correcta una serie de preguntas.
- **Service:** En *Android*, un servicio es un componente que puede realizar tareas de larga duración, y que no proporciona interfaz de usuario.
- **Slide:** En *Android*, es el acto de arrastrar un dedo en la pantalla de forma lateral.
- **Salt:** En criptografía, es un dato aleatorio que se usa como entrada adicional en una función de una vía que cifra una contraseña. Es utilizada para defenderse de ataques de diccionario.
- **RainbowTable:** Son tablas de consulta que ofrecen un compromiso entre tiempo y espacio para obtener claves en texto simple a partir del resultado de una función *hash*.
- **ViewPager:** En *Android*, es un componente que permite al usuario cambiar de página mediante el uso de *Slides*, la información que va en las páginas es cargada al *ViewPager* gracias a un adaptador.

REFERENCIAS

- [1]https://www.noellevitz.com/documents/shared/Papers_and_Research/2013/2013_E-Expectations_mobile.pdf
- [2] <http://www.duolingo.com/>
- [3] <http://www.triviados.net/>
- [4] <http://www.eclipse.org/>
- [5] <http://developer.android.com/about/index.html>
- [6] <http://php.net/>
- [7] <http://www.postgresql.org/>
- [8] <http://phpPgadmin.sourceforge.net/doku.php>
- [9] Análisis y Diseño de Sistemas, 6ta Edición – Kenneth E. Kendall & Julie E. Kendall pag 151
- [10]<http://www.veracode.com/security/man-in-the-middle-attack>
- [11]<http://security.stackexchange.com/questions/14025/why-is-using-salt-more-secure>
- [12] <http://keatas.kuliukas.com/RainbowTables/>

- [13]<http://developer.android.com/guide/components/fragments.html>
- [14]<http://developer.android.com/reference/android/support/v4/view/ViewPager.html>
- [15]<http://developer.android.com/guide/topics/ui/actionbar.html>
- [16] <http://developer.android.com/guide/topics/ui/menus.html>
- [17]<http://actionbarsherlock.com/>
- [18]http://en.wikipedia.org/wiki/Elo_rating_system

